(19)日本国特許庁（ＪＰ） (12) 公開特許公報（A） (11)特許出願公開番号

特開平4－321136

(43)公開日 平成４年(1992)11月11日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ 9/46	３４０ Ｂ	8120－5Ｂ		

審査請求 未請求 請求項の数２（全 ６ 頁）

(21)出願番号 特願平3－49316

(22)出願日 平成３年(1991)３月14日

(71)出願人 000005223
富士通株式会社
神奈川県川崎市中原区上小田中1015番地

(72)発明者 芳賀 豊
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

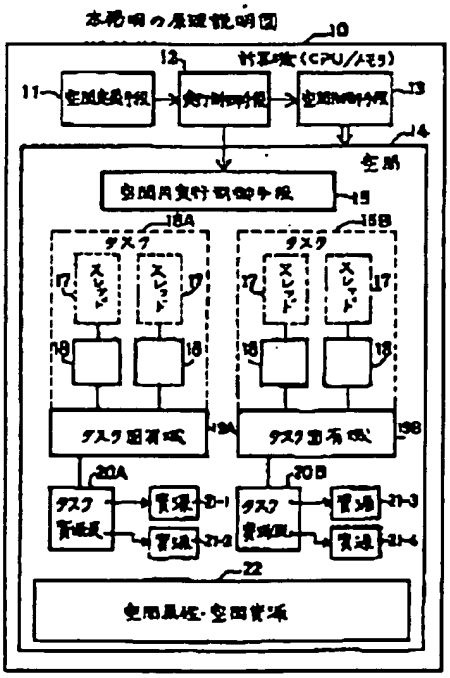
(74)代理人 弁理士 小笠原 吉教 （外２名）

(54)【発明の名称】 マルチタスク制御方法および制御装置

(57)【要約】

【目的】計算機システムにおいて，それぞれに単独のプログラム実行環境を前提として作成されたプログラムを同一の空間内で同時に複数実行させるマルチタスク制御方法および制御装置に関し，同一空間内での多重処理を容易に実現し，実行性能を向上させることを目的とする。

【構成】外部変数やファイル資源などを共有するスレッド17をグループ化し，それぞれタスク16A，16B と呼ぶ制御単位により管理する。このタスクを同一空間14内に複数個生成し，異なるタスクにおけるスレッド間相互の影響をなくした動作環境を構築する。空間定義手段11によって，マルチタスク空間の作成を指定すると，対象空間14において空間内実行制御手段15がそのマルチタスクの環境を整える。



1

【特許請求の範囲】

【請求項1】 計算機システムにおける仮想空間内でプログラムを実行させるCPU実行体を複数個走行させるマルチタスク制御方法において，一つの空間内で，それぞれCPU実行体としてCPU実行権を得てプログラムを実行する複数のスレッド(17)と，これらのスレッド群で共用されるデータ領域およびこれらのスレッド群にその他の計算機資源を割り当てるための資源表とを一組とする制御単位を，複数個生成し，前記制御単位のそれぞれに単独のプログラム実行環境を前提として作成された同種または異種のプログラムを，同一空間内で同時に複数個走行させ，同一空間内での複数のスレッド群による多重処理を行うことを特徴とするマルチタスク制御方法。

【請求項2】 計算機システムにおける仮想空間内でプログラムを実行させるCPU実行体を複数個走行させるマルチタスク制御装置において，一つの空間に対して，CPU実行体としてCPU実行権を得てプログラムを実行する1または複数のスレッド(17)を含む制御単位であって，それらのスレッド群が所定の範囲内の計算機資源を共用する単位となる制御単位を複数設けることを指示する空間定義手段(11)と，複数の制御単位を設けることを指示する空間定義が行われた場合に，作成された対象空間内で前記複数の制御単位の環境を作成し，それぞれの制御単位ごとにプログラムのアドレス解決を行ってプログラムを実行させる空間内実行制御手段(15)とを備え，各制御単位内では，複数のスレッド(17)が計算機資源を共用し，異なる制御単位で動作するスレッド間では，各制御単位に割り当てられた資源を個別に扱うようにしたことを特徴とするマルチタスク制御装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は，計算機システムにおいて，それぞれに単独のプログラム実行環境を前提として作成されたプログラムを同一の空間内で同時に複数実行させるマルチタスク制御方法および制御装置に関する。

【0002】近年のオペレーティングシステムでは，並列プログラミングや，サーバ空間等における多重処理の要求に伴い，空間内に複数のCPU実行体を生成する機能が要求されている。このため，軽量なCPU実行体であるスレッドが提供されているが，各スレッドは，個々のスレッドが一時的な作業領域として使用するスタック領域を除くと，すべてがスレッド間共用資源となるため，常にスレッド間の相互の影響を考慮してプログラミングする必要がある。

【0003】同一空間内での多重処理を容易に実現し，実行性能を向上させるためには，同一空間内のスレッド間の相互作用を改善する技術が望まれる。

【0004】

【従来の技術】従来のマルチスレッド方式では，空間内の資源をすべてのスレッドが共有して動作していた。なお，スレッドとは，CPU実行体としてCPU実行権を得るプログラムの実行単位であって，ハードウェア・レジスタ，スタック等のプログラム実行環境を切り　える単位となるものである。

2

【0005】ところが，大域的なデータ領域を含む空間内の共用資源は，空間を範囲としてすべてのスレッドによって操作され得るため，各スレッドは，資源の操作に際して，他スレッドとの排他および資源操作に伴う副作用，すなわちスレッド間の相互作用を常に考慮しなければならない。また，本来単独のプログラム実行環境を前提として作成されたプログラム（C言語プログラムではmain()関数を持つプログラムなど）を実行する場合には，プログラム間の相互作用を避けるため，別の空間で実行する必要がある。

【0006】また，従来のオペレーティングシステムによるタスク制御では，CPU実行権を持つ単位とプログラムおよび資源とが一対一に結合して動作するため，ある資源を持つ一つのプログラムについて，複数のCPU実行体で並列処理することはできなかった。

【0007】

【発明が解決しようとする課題】従来のマルチスレッド方式では，多重処理を行うためには，常にスレッド間の相互作用を考慮してプログラミングしなければならない問題点を生じていた。また，単独のプログラム実行環境を要求するプログラム（C言語プログラムではmain()関数を持つプログラム）として作成した場合には，これらのプログラムを同一空間内で複数実行させることはできず，このようなプログラムを複数実行させるためには，必ず別の空間を作成して実行しなければならないため，空間作成のオーバヘッドが大きくなるという問題があった。

【0008】一般的には，同一空間内で，それぞれにmain()関数を持つ全く独立なプログラムを複数実行させることは，プログラム実行上のセキュリティの問題および空間に唯一割り当てられる空間属性および空間資源の競合から許可できない。しかしながら，プログラム実行上のセキュリティ条件に反せず，空間に唯一の空間属性および空間資源に競合が生じない範囲においては，これを許可することが可能である。

【0009】本発明は上記問題点の解決を図り，同一空間内にmain()関数を持つプログラムを実行する実行体（タスク）を複数生成することにより，同一空間内での多重処理の実現を容易にし，かつ少ない資源での複数の実行体の実行を可能とし，実行性能を向上させることを目的とする。

【0010】

【課題を解決するための手段】図1は，本発明の原理説明図である。図中，10はCPUおよびメモリからなる計算機，11は同一空間に複数のタスクを生成すること

3

を指示することができる空間定義手段．１２はオペレーティングシステムによる実行制御手段．１３は空間を作成し管理する空間制御手段，１４は計算機システムにおいてプログラムが動作するための仮想的な空間．１５は空間内実行制御手段．１６Ａ，１６Ｂは割り当てられた資源を複数のスレッドが共有する制御単位となるタスク，１７はＣＰＵ実行権を得るプログラムの実行単位であるスレッド．１８はスタック領域およびスレッド固有域であり，個々のスレッドで局所的なデータ領域として使用されるもの，１９Ａ，１９Ｂはタスク内のスレッドで共用される大域データ領域であるタスク固有域．２０Ａ，２０Ｂはタスク資源表であり，それぞれのタスクに割り当てられた資源を管理するための制御表，２１－１ないし２１－４はファイルオープン環境やメモリの動的獲得領域というようなプログラムが使用する計算機の資源．２２は空間属性および空間資源であり，個々のタスクで占有されない資源を表す．

【００１１】タスク１６Ａ，１６Ｂは，それぞれ複数のスレッド１７と，各スレッド群で共用されるタスク固有域１９Ａ，１９Ｂおよび資源を管理するタスク資源表２０Ａ，２０Ｂを一組とする制御単位である．本発明では，同一空間１４内に，これらのタスクを複数個生成する．

【００１２】そして，これらの各タスク１６Ａ，１６Ｂのそれぞれに単数のプログラム実行環境を前提として作成された同種または異種のプログラムを，同一空間１４内で同時に複数個走行させ，同一空間１４内での複数のスレッド群による多重処理を行う．

【００１３】空間定義手段１１は，一つの空間１４を作成し，その空間１４に対して，複数のタスク１６Ａ，１６Ｂ等を設けることを定義し指示する手段である．空間定義手段１１によるマルチタスクの指示により，実行制御手段１２は，空間制御手段１３に対し空間１４の作成を依頼し，空間制御手段１３は，空間１４を作成する．この空間の作成方法については従来技術と同様である．

【００１４】実行制御手段１２は，空間１４が作成されたならば，その空間１４上で空間内実行制御手段１５を起動する．空間内実行制御手段１５は，マルチタスクの指示がある場合には，タスク固有域１９Ａ，１９Ｂ，タスク資源表２０Ａ，２０Ｂなどの複数の制御単位の環境を作成し，それぞれの制御単位ごとにプログラムのアドレス解決を行ってプログラムを実行させる．

【００１５】各制御単位内，例えばタスク１６Ａ内では，複数のスレッド１７がタスク固有域１９Ａおよびタスク資源表２０Ａによって管理される資源２１－１，２１－２を共用し，異なる制御単位で動作するスレッド間，すなわちタスク１６Ａのスレッド１７とタスク１６Ｂのスレッド１７間では，各制御単位に割り当てられた資源を相互に影響させず個別に扱うようになっている．

【００１６】

4

【作用】本発明では，図１に示すように，空間１４内のそれぞれのタスク１６Ａ，１６Ｂは，そのタスク内のスレッド１７でのみ共用されるタスク固有域１９Ａ，１９Ｂと，各タスクに割り当てられる資源を管理するためのタスク資源表２０Ａ，２０Ｂを持ち，プログラム実行に係わる資源を，各タスク１６Ａ，１６Ｂごとに占有してプログラムを実行するようにしている．空間属性・空間資源２２は，実行体としての空間を補完する．これらは空間１４で唯一のものでありタスク１６Ａ，１６Ｂで共用されるが，競合が生じないように使用が制限される．

【００１７】したがって，プログラム実行上のセキュリティ条件に反せず，空間１４で唯一の空間属性・空間資源２２に競合が生じない範囲においては，同一空間１４内でｍａｉｎ（）関数を持つプログラムを複数実行することが可能となる．

【００１８】

【実施例】図２は本発明の一実施例によるタスク構成例を示す図，図３は本発明の一実施例に係る実行制御の構成例を示すブロック図，図４は本発明の一実施例による空間の作成からタスク作成までの流れ図，図５は本発明の一実施例によるタスク作成の流れ図である．

【００１９】図２は，例えばＣ言語によるｍａｉｎ（）関数を持つプログラムが要求する資源の各実行体への配分およびその管理方法を示している．スレッド１７は，従来技術と同様に，それぞれスタック領域およびスレッド固有域１８を持つ．このスタック領域およびスレッド固有域１８は，各スレッド１７で動作するプログラムが使用するスタックと局所変数の格納域等に使用される．

【００２０】タスク１６は，これらのスレッド１７とその資源とをグループ化したものである．各タスク１６ごとに設けられるタスク固有域１９は，そのタスク１６に属するスレッド１７が共用する外部変数の領域などに使用される．また，タスク資源表２０が各タスク１６対応に用意され，そのタスク１６が占有する資源２１－１，２１－２などを管理する．このタスク１６を空間１４内に複数個作成することにより，異なるタスク１６上で動作するスレッド１７は，それぞれ他のタスク１６が持つ資源を意識することなく処理できるようになる．

【００２１】空間属性・空間資源２２は，例えば空間を識別する空間ｉｄまたはプロセスｉｄとか，各種資源の使用権限を示すユーザｉｄやグループｉｄなどの各空間に固有の管理情報などからなる資源である．空間属性・空間資源２２は，各タスク１６で共用される．

【００２２】図２に示すタスク構成を実現するための環境は，図３に示すようになっている．図３における実線の矢印は，一つの空間内に一つのタスク（スレッドは複数）が存在するシングルタスク制御の流れであって，従来技術に相当するものの流れを示している．点線の矢印は，本発明に係るマルチタスク制御の流れを示している．

【0023】マルチタスク制御を行う場合，プログラム名，タスク多重度（タスク数），例えば必要メモリ量などのプログラム実行環境情報を定義した空間定義３０を，マルチタスク実行制御空間３３に対して送る。マルチタスク実行制御空間３３は，空間スーパバイザ空間３４に対し，空間の作成依頼を行う。

【0024】空間スーパバイザ空間３４では，空間管理部３５から空間制御空間３６を起動し，空間制御空間３６の仮想空間作成部３７で対象空間４０を作成し，空間初期化部３８で作成した対象空間４０の初期化を行う。この空間の作成処理は，シングルタスクの場合もマルチタスクの場合も同様である。

【0025】対象空間４０が作成されたならば，マルチタスク実行制御空間３３は，対象空間４０に対して実行制御情報の送信を行う。このときの入力情報３９は，シングルタスクであるかマルチタスクであるかの実行制御種別，プログラム名，マルチタスク種別である場合にタスク多重度およびプログラム実行環境情報などである。

【0026】対象空間４０では，空間内実行制御部４１が受信した実行制御情報を解析し，マルチタスクの指定がある場合には，マルチタスク実行制御部４３によってタスク制御部４４へのタスク生成依頼を，タスク多重度の数だけ行う。これにより，タスク制御部４４によって図２に示すタスク１６の環境が複数個作成される。

【0027】シングルタスク制御では，要求元空間３１におけるシングルタスク実行制御部３２が，空間の作成依頼を空間スーパバイザ空間３４に行い，同様に対象空間４０を作成させる。次に，シングルタスクの実行制御種別とプログラム名とプログラム実行環境情報とを対象空間４０に送信し，対象空間４０の空間内実行制御部４１は，シングルタスク実行制御部４２により，１タスクの生成をタスク制御部４４に依頼する。このシングルタスク制御は，従来技術によるマルチスレッド方式の制御に相当している。

【0028】次に，図４に示す処理①～⑨に従って，空間の作成からタスク作成までの処理の流れを説明する。

① マルチタスク実行制御空間３３または要求元空間３１の実行制御部から，空間の作成依頼を行う。

② これにより，物理的な仮想アドレス空間の作成処理を行う。

③ そして，作成した仮想アドレス空間を論理的な実行体として構築するための空間制御表および空間資源の作成と初期化を行う。

④ 空間が作成されたならば，空間作成処理を依頼した実行制御部が，実行プログラム情報および対象空間内でのプログラム実行環境を確定する実行制御情報を，空間内実行制御部４１に送信する。

【0029】⑤ 空間内実行制御部４１は，処理④で送信された実行制御情報をもとに，プログラム実行環境を設定する処理を行う。

⑥ シングルタスク空間の作成時には，シングルタスク実行制御部４２がタスクの生成をタスク制御部４４に依頼し，プログラムの読込み，プログラムのアドレス解決を行って，そのプログラムを実行させる。

【0030】⑦ マルチタスク空間の作成時には，マルチタスク実行制御部４３が，タスクを管理するための自身のマルチタスク制御環境を構築する。

⑧ そして，各タスクで共通に実行されるプログラムを事前に読み込んで，対象空間４０にバインドする。

⑨ 次に，タスク制御部４４に対し，タスクの生成を依頼し，各タスクごとにプログラムのアドレス解決およびプログラムの実行を制御する。その処理を，指定個数のタスクを作成するまで繰り返す。

【0031】図５は，さらに詳しいタスク作成の流れ図を示している。以下，図５に示す処理①～⑦に従って説明する。

① まず，タスクの実行情報を管理するためのタスク制御表（ＴＭＢ）を作成する。なお，制御表内部の構造については，本発明の実施にあたって種々の設計が可能であり，本発明の要旨から外れるので詳しい説明を省略する。

② 次に，タスク上で動作するプログラムが獲得する資源を，該タスクに保持して管理するためのタスク資源表（ＲＬＳＴ）を作成する。

③ さらに，タスク内で動作する最初のスレッド（初期スレッドという）の実行情報を管理するためのスレッド制御表（ＴＨＣＴ）を作成する。管理するスレッド資源が多岐にわたらないため，ここでは，スレッド制御表がスレッド資源表の役目も兼ねている。

④ 初期スレッドのためのスタック領域の獲得を行う。

⑤ シングルタスク空間の作成時には，シングルタスク実行制御部４２により，指定されたプログラム情報をもとに，プログラムを読み込んで，対象空間４０にバインドし，アドレス解決処理を行う。アドレス解決処理により，プログラムが要求する作業領域があれば，その作業領域を獲得・設定した上で，タスク資源表またはスレッド資源表に登録する。

⑥ 一方，マルチタスク空間の作成時には，そのタスク上で，マルチタスク実行制御部４３のアドレス解決処理部を呼び出す。このアドレス解決処理では，既にプログラムが空間にバインド済みであることから，アドレスの解決のみを行い，その実行開始アドレスを通知する。

⑦ 処理⑤または処理⑥のアドレス解決処理によって得られた実行開始アドレスより，プログラムの実行を開始する処理を行う。開始した初期スレッドのプログラムから，必要に応じて他のスレッドを生成する処理を起動することにより，そのタスク上で複数のスレッドが動作することになる。

【0032】

【発明の効果】以上説明したように，本発明によれば，

7

同一空間内で，例えばC言語プログラムにおけるｍａｉｎ（）関数を持つような，単独のプログラム実行環境を要求するプログラムを，複数同時に実行させることができるようになり，空間内における多重処理の容易な実現が可能になるとともに，少ない資源での複数の実行体による実行が可能になる。したがって，プログラム開発の容易化および実行性能の向上に寄与するところが大きい。

【図面の簡単な説明】

【図１】本発明の原理説明図である。

【図２】本発明の一実施例によるタスク構成例を示す図である。

【図３】本発明の一実施例に係る実行制御の構成例を示すブロック図である。

【図４】本発明の一実施例による空間の作成からタスク作成までの流れ図である。

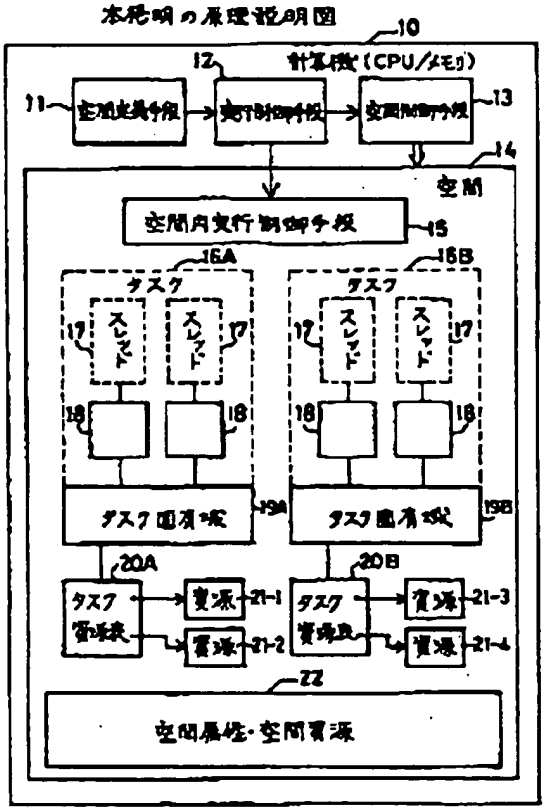
8

【図５】本発明の一実施例によるタスク作成の流れ図である。

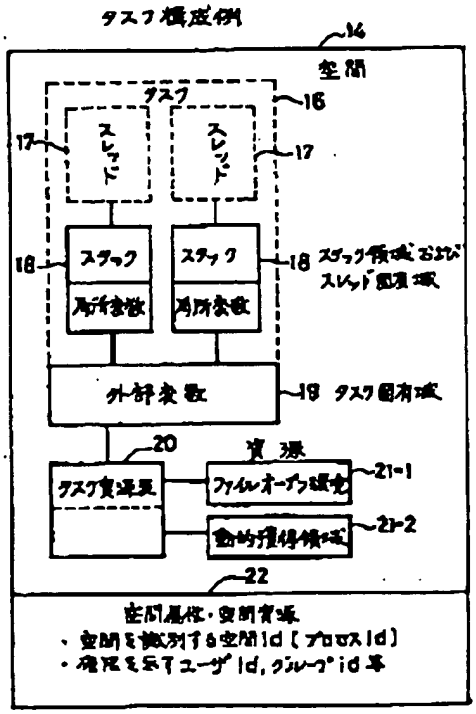
【符号の説明】

１０　計算機
１１　空間定義手段
１２　実行制御手段
１３　空間制御手段
１４　空間
１５　空間内実行制御手段
１６Ａ，１６Ｂ　タスク
１７　スレッド
１８　スタック領域およびスレッド固有域
１９Ａ，１９Ｂ　タスク固有域
２０Ａ，２０Ｂ　タスク資源表
２１－１～２１－４　資源
２２　空間属性・空間資源

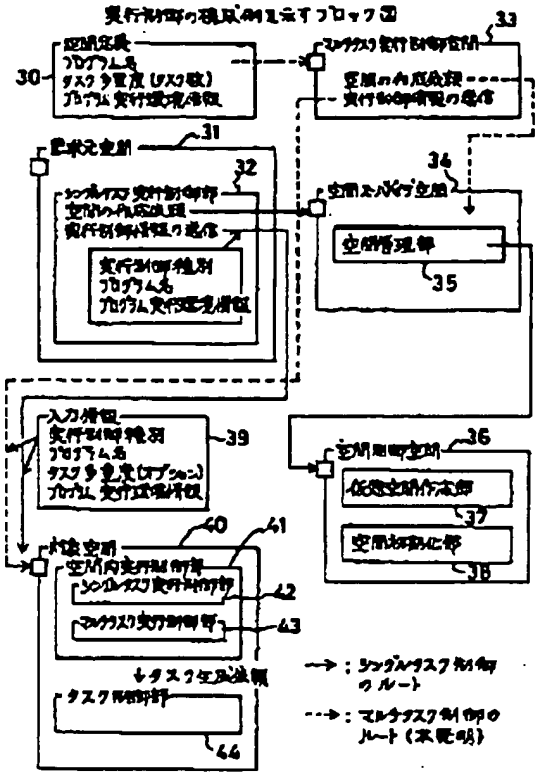
【図１】



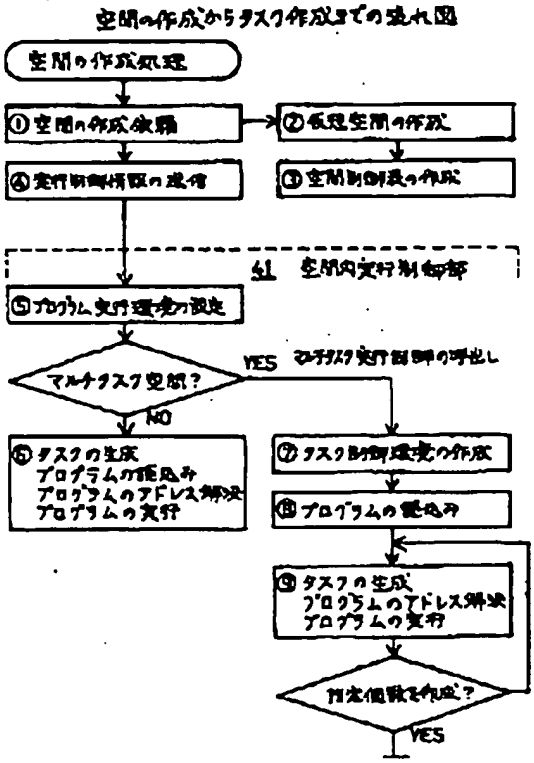
【図２】



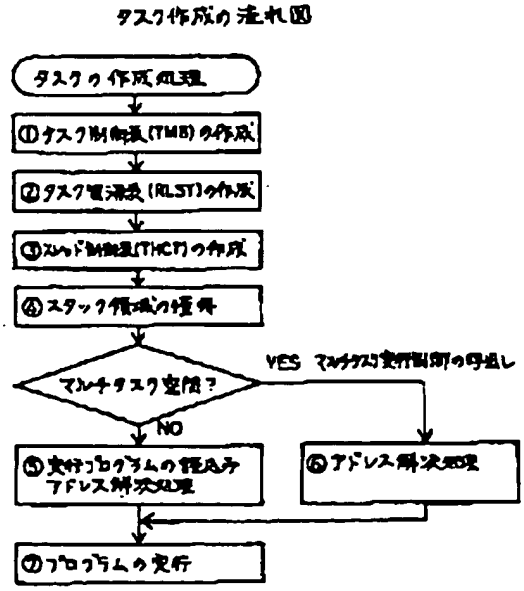
【図3】


実行制御の構成例を示すブロック図
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
→：シングルタスク制御のルート
⇢：マルチタスク制御のルート（本発明）

【図4】


空間の作成からタスク作成までの流れ図
空間の作成処理
①空間の作成依頼
②仮想空間の作成
④実行制御情報の通信
③空間制御表の作成
41 空間内実行制御部
⑤プログラム実行環境の設定
マルチタスク空間？
YES
マルチタスク実行制御の呼出し
NO
⑥タスクの生成
プログラムの読込み
プログラムのアドレス解決
プログラムの実行
⑦タスク制御環境の作成
⑧プログラムの読込み
⑨タスクの生成
プログラムのアドレス解決
プログラムの実行
指定個数を作成？
YES

【図5】


タスク作成の流れ図
タスクの作成処理
①タスク制御表(TMB)の作成
②タスク管理表(RLST)の作成
③スレッド制御表(THCT)の作成
④スタック領域の確保
マルチタスク空間？
YES
マルチタスク実行制御の呼出し
NO
⑤実行プログラムの読込み
アドレス解決処理
⑥アドレス解決処理
⑦プログラムの実行